

```
#include <arpa/inet.h>
#include <netdb.h>
#include <netinet/in.h>
#include <stdint.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ifaddrs.h>
#include <unistd.h>
```

```
#include "net.h"
#include "transport.h"
#include "esp.h"
```

```
#define DEBUG
```

```
uint16_t cal_ipv4_cksm(struct iphdr* iphdr)
{
    // [TODO]: Finish IP checksum calculation
    iphdr->check = 0;
    unsigned short *addr = (unsigned short *)iphdr;
    unsigned int count = iphdr->ihl<<2;
    register unsigned long sum = 0;
    while ( count > 1)
    {
        sum += *addr++;
        count -= 2;
    }
    if (count > 0)
    {
```

```

        sum += ((*addr)&htons(0xFF00));
    }
    while ( sum >> 16)
    {
        sum = (sum & 0xffff) + (sum >> 16);
    }
    sum = ~sum;
    iphdr->check = (uint16_t)sum;
    return ((uint16_t)sum);
}

uint8_t *dissect_ip(Net *self, uint8_t *pkt, size_t pkt_len)
{
    // [TODO]: Collect information from pkt.
    // Return payload of network layer
    struct sockaddr_in source,dest;
    struct iphdr *ip = (struct iphdr *)pkt;
    memcpy(&self->ip4hdr, ip, sizeof(struct iphdr));
    // set hdrlen
    self->hdrlen = (size_t)ip->ihl<<2;
    // set plen
    self->plen = pkt_len - self->hdrlen;
    // set pro
    switch (ip->protocol)
    {
    case ESP:
        self->pro = ESP;
        break;
    case IPv4:
        self->pro = IPv4;

```

```

        break;
    case TCP:
        self->pro = TCP;
        break;
    default:
        self->pro = UNKN_PROTO;
        break;
}

memset(&source, 0, sizeof(source));
source.sin_addr.s_addr = ip->saddr;
memset(&dest, 0, sizeof(dest));
dest.sin_addr.s_addr = ip->daddr;
// set source & dest IP

strcpy(self->src_ip,inet_ntoa(source.sin_addr));
strcpy(self->dst_ip,inet_ntoa(dest.sin_addr));

#ifdef DEBUG
printf("\nIP Header\n");
printf("\t|-Version          : %d\n",(unsigned int)ip->version);
printf("\t|-Internet Header Length : %d DWORDS or %d Bytes\n",(unsigned int)ip->ihl,((unsigned int)(ip->ihl))*4);
printf("\t|-Type Of Service   : %d\n",(unsigned int)ip->tos);
printf("\t|-Total Length      : %d Bytes\n",ntohs(ip->tot_len));
printf("\t|-Identification   : %d\n",ntohs(ip->id));
printf("\t|-Time To Live       : %d\n",(unsigned int)ip->ttl);
printf("\t|-Protocol          : %d\n",(unsigned int)ip->protocol);
printf("\t|-Header Checksum    : %d\n",ntohs(ip->check));
printf("\t|-Source IP         : %s\n", inet_ntoa(source.sin_addr));
printf("\t|-Destination IP    : %s\n",inet_ntoa(dest.sin_addr));

```

```

printf("!my checksum: %d\n",ntohs(cal_ipv4_cksm(ip)));
printf("IP pkt srcIP: %s\n", self->src_ip);
printf("IP pkt dstIP: %s\n", self->dst_ip);
printf("IP pkt protocol: %d\n", self->pro);
printf("self->ip4hdr.Protocol: %d\n", (unsigned int)self->ip4hdr.protocol);
#endif

return pkt + self->hdrlen;
}

```

```

Net *fmt_net_rep(Net *self)
{
    // [TODO]: Fill up self->ip4hdr (prepare to send)

    // struct iphdr
    // {
    // #if __BYTE_ORDER == __LITTLE_ENDIAN
    //     unsigned int ihl:4;
    //     unsigned int version:4;
    // #elif __BYTE_ORDER == __BIG_ENDIAN
    //     unsigned int version:4;
    //     unsigned int ihl:4;
    // #else
    // # error "Please fix <bits/endian.h>"
    // #endif
    //     uint8_t tos;
    //     uint16_t tot_len;
    //     uint16_t id;
    //     uint16_t frag_off;
    //     uint8_t ttl;

```

```

//  uint8_t protocol;
//  uint16_t check;
//  uint32_t saddr;
//  uint32_t daddr;
//  /*The options start here. */
// };

return self;
}

void init_net(Net *self)
{
    if (!self) {
        fprintf(stderr, "Invalid arguments of %s.", __func__);
        exit(EXIT_FAILURE);
    }

    self->src_ip = (char *)malloc(INET_ADDRSTRLEN * sizeof(char));
    self->dst_ip = (char *)malloc(INET_ADDRSTRLEN * sizeof(char));
    self->x_src_ip = (char *)malloc(INET_ADDRSTRLEN * sizeof(char));
    self->x_dst_ip = (char *)malloc(INET_ADDRSTRLEN * sizeof(char));
    self->hdrlen = sizeof(struct iphdr);

    self->dissect = dissect_ip;
    self->fmt_rep = fmt_net_rep;
}

```

	/*
	* CIS 0000
	* CIS 00000000 1000 CIS 00 2000 CIS 00 3000 CIS 0 0/00 400000000000 50CIS 00000000

	/*	
	* CIS 接口	
	* CIS 接口规范 1.0.0 CIS 接口 2.0.0 CIS 接口 3.0.0 CIS 接口/规范 4.0.0.0.0.0 接口 5.0.0 CIS 接口规范	
	*	
	* OpenAPI spec version: v1	
	*	
	*	
	* NOTE: This class is auto generated by the swagger code generator program.	
	* https://github.com/swagger-api/swagger-codegen.git	
	* Do not edit the class manually.	
	*/	
	package com.huawei.cloudcampus.api.model;	
	import java.util.Objects;	
	import com.google.gson.TypeAdapter;	
	import com.google.gson.annotations.JsonAdapter;	
	import com.google.gson.annotations.SerializedName;	
	import com.google.gson.stream;	

	am.JsonReader;	
	import com.google.gson.stre am.JsonWriter;	
	import io.swagger.annotation s.ApiModel;	
	import io.swagger.annotation s.ApiModelProperty;	
	import java.io.IOException;	
	import java.util.ArrayList;	
	import java.util.List;	
	/**	
	* InterdictionDTO	
	*/	
	@javax.annotation.Ge nerated(value = "io.swagger.codegen.l anguages.JavaSdnClie ntCodegen", date = "2019-12- 17T15:17:59.469+08: 00")	
	public class InterdictionDTO {	
	@SerializedName("bl ockId")	
	private String blockId = null;	
	@SerializedName("te nant")	
	private String tenant = null;	
	@SerializedName("pr oducer")	
	private String	

	producer = null;	
	@SerializedName("direction")	
	private Integer direction = null;	
	@SerializedName("srcIps")	
	private List<String> srcIps = null;	
	@SerializedName("dstIps")	
	private List<String> dstIps = null;	
	public InterdictionDTO blockId(String blockId) {	
	this.blockId = blockId;	
	return this;	
	}	
	/**	
	* 返回ID为UUID的拦截记录	
	* return blockId	
	*/	
	@ApiModelProperty(required = true, value = "返回ID为UUID的拦截记录")	
	public String getBlockId() {	
	return blockId;	
	}	
	/**	
	* 返回ID为UUID的拦截记录	
	* Param blockId	
	*/	

	public void setBlockId(String blockId) {	
	this.blockId = blockId;	
	}	
	public InterdictionDTO tenant(String tenant) {	
	this.tenant = tenant;	
	return this;	
	}	
	/**	
	* ID UUID	
	* return tenant	
	**/	
	@ApiModelProperty(v alue = "ID UUID ")	
	public String getTenant() {	
	return tenant;	
	}	
	/**	
	* ID UUID	
	* Param tenant	
	**/	
	public void setTenant(String tenant) {	
	this.tenant = tenant;	
	}	
	public InterdictionDTO producer(String producer) {	
	this.producer = producer;	

	return this;	
	}	
	/**	
	* 生产者	
	* return producer	
	**/	
	@ApiModelProperty(v alue = "生产者")	
	public String getProducer() {	
	return producer;	
	}	
	/**	
	* 生产者	
	* Param producer	
	**/	
	public void setProducer(String producer) {	
	this.producer = producer;	
	}	
	public InterdictionDTO direction(Integer direction) {	
	this.direction = direction;	
	return this;	
	}	
	/**	
	* 0-1-生产者 0	
	* return direction	
	**/	
	@ApiModelProperty(r equired = true, value = "0-1-生产者")	

	00000")	
	public Integer	
	getDirection() {	
	return direction;	
	}	
	/**	
	* 000000-00001-000000	
	00000	
	* Param direction	
	**/	
	public void	
	setDirection(Integer	
	direction) {	
	this.direction =	
	direction;	
	}	
	public	
	InterdictionDTO	
	srcIps(List<String>	
	srcIps) {	
	this.srcIps = srcIps;	
	return this;	
	}	
	public	
	InterdictionDTO	
	addSrcIpsItem(String	
	srcIpsItem) {	
	if (this.srcIps == null)	
	{	
	this.srcIps = new	
	ArrayList<String>();	
	}	
	this.srcIps.add(srcIpsI	
	tem);	
	return this;	
	}	
	/**	
	* IP 0000000000000000	

	IPv4IPv6 8 IP IP IP IP IP	
	* return srcIps	
	**/	
	@ApiModelProperty(v alue = " IP IPv4IPv6 8 IP IP IP")	
	public List<String>	
	getSrcIps() {	
	return srcIps;	
	}	
	/**	
	* IP IPv4IPv6 8 IP IP IP	
	* Param srcIps	
	**/	
	public void	
	setSrcIps(List<String	
	> srcIps) {	
	this.srcIps = srcIps;	
	}	
	public	
	InterdictionDTO	
	dstIps(List<String>	
	dstIps) {	
	this.dstIps = dstIps;	
	return this;	
	}	
	public	
	InterdictionDTO	
	addDstIpsItem(String	
	dstIpsItem) {	
	if (this.dstIps == null)	
	{	

	this.dstlps = new ArrayList<String>();	
	}	
	this.dstlps.add(dstlpsI tem);	
	return this;	
	}	
	/**	
	* 目标 IP 地址列表 支持 IPV4、IPV6 地址 最多 1 万个 IP 地址 IP 地址列表	
	* return dstlps	
	**/	
	@ApiModelProperty(v alue = "目标 IP 地址 列表支持 IPV4、IPV6 地址最多 1 万个 IP 地址 IP 地址列表")	
	public List<String> getDstlps() {	
	return dstlps;	
	}	
	/**	
	* 目标 IP 地址列表 支持 IPV4、IPV6 地址 最多 1 万个 IP 地址 IP 地址列表	
	* Param dstlps	
	**/	
	public void setDstlps(List<String > dstlps) {	
	this.dstlps = dstlps;	
	}	
	@Override	
	public boolean equals(java.lang.Obj	

	ct o) {	
	if (this == o) {	
	return true;	
	}	
	if (o == null getClass() != o.getClass()) {	
	return false;	
	}	
	InterdictionDTO interdictionDTO = (InterdictionDTO) o;	
	return Objects.equals(this.bl ockId, interdictionDTO.blockI d) &&	
	Objects.equals(this.te nant, interdictionDTO.tenan t) &&	
	Objects.equals(this.pr oducer, interdictionDTO.produ cer) &&	
	Objects.equals(this.di rection, interdictionDTO.direct ion) &&	
	Objects.equals(this.sr clps, interdictionDTO.srclps) &&	
	Objects.equals(this.ds tlps, interdictionDTO.dstlps);	
	}	
	@Override	
	public int hashCode() {	

	return	
	Objects.hash(blockId,	
	tenant, producer,	
	direction, srcIps,	
	dstIps);	
	}	
	@Override	
	public String	
	toString() {	
	StringBuilder sb =	
	new StringBuilder();	
	sb.append("class	
	InterdictionDTO {\n");	
	sb.append(" blockId:	
	").append(toIndented	
	String(blockId)).appen	
	d("\n");	
	sb.append(" tenant:	
	").append(toIndented	
	String(tenant)).appen	
	d("\n");	
	sb.append(" producer:	
	").append(toIndented	
	String(producer)).app	
	end("\n");	
	sb.append(" direction:	
	").append(toIndented	
	String(direction)).app	
	end("\n");	
	sb.append(" srcIps:	
	").append(toIndented	
	String(srcIps)).append	
	(" \n");	
	sb.append(" dstIps:	
	").append(toIndented	
	String(dstIps)).appen	
	d("\n");	
	sb.append("}");	

	return sb.toString();	
	}	
	/**	
	* Convert the given object to string with each line indented by 4 spaces	
	* (except the first line).	
	*/	
	private String toIndentedString(java .lang.Object o) {	
	if (o == null) {	
	return "null";	
	}	
	return o.toString().replace("\n", "\n ");	
	}	
	}	
Footer		
	*	
	* OpenAPI spec version: v1	
	*	
	*	
	* NOTE: This class is auto generated by the swagger code generator program.	
	* https://github.com/swagger-api/swagger-codegen.git	
	* Do not edit the class manually.	
	*/	
	package com.huawei.cloudcampus.api.model;	
	import java.util.Objects;	

	import com.google.gson.TypeAdapter;
	import
	com.google.gson.annotations.JsonAdapter;
	import
	com.google.gson.annotations.SerializedName;
	import com.google.gson.stream.JsonReader;
	import com.google.gson.stream.JsonWriter;
	import io.swagger.annotations.ApiModel;
	import
	io.swagger.annotations.ApiModelProperty;
	import java.io.IOException;
	import java.util.ArrayList;
	import java.util.List;
	/**
	* InterdictionDTO
	*/
	@javax.annotation.Generated(value =
	"io.swagger.codegen.languages.JavaSdnClientCo
	degen", date = "2019-12-
	17T15:17:59.469+08:00")
	public class InterdictionDTO {
	@SerializedName("blockId")
	private String blockId = null;
	@SerializedName("tenant")
	private String tenant = null;
	@SerializedName("producer")
	private String producer = null;
	@SerializedName("direction")
	private Integer direction = null;
	@SerializedName("srcIps")
	private List<String> srcIps = null;
	@SerializedName("dstIps")
	private List<String> dstIps = null;
	public InterdictionDTO blockId(String blockId) {
	this.blockId = blockId;

	return this;
	}
	/**
	* ID UUID
	* return blockId
	**/
	@ApiModelProperty(required = true, value = "ID UUID")
	public String getBlockId() {
	return blockId;
	}
	/**
	* ID UUID
	* Param blockId
	**/
	public void setBlockId(String blockId) {
	this.blockId = blockId;
	}
	public InterdictionDTO tenant(String tenant) {
	this.tenant = tenant;
	return this;
	}
	/**
	* ID UUID
	* return tenant
	**/
	@ApiModelProperty(value = "ID UUID")
	public String getTenant() {
	return tenant;
	}
	/**
	* ID UUID
	* Param tenant
	**/
	public void setTenant(String tenant) {
	this.tenant = tenant;

	}
	public InterdictionDTO producer(String producer)
	{
	this.producer = producer;
	return this;
	}
	/**
	* 生产者
	* return producer
	**/
	@ApiModelProperty(value = "生产者")
	public String getProducer() {
	return producer;
	}
	/**
	* 生产者
	* Param producer
	**/
	public void setProducer(String producer) {
	this.producer = producer;
	}
	public InterdictionDTO direction(Integer
	direction) {
	this.direction = direction;
	return this;
	}
	/**
	* 方向0-1-方向 0
	* return direction
	**/
	@ApiModelProperty(required = true, value = "方向0-1-方向 0")
	public Integer getDirection() {
	return direction;
	}
	/**

	public InterdictionDTO dstlps(List<String> dstlps) {
	this.dstlps = dstlps;
	return this;
	}
	public InterdictionDTO addDstlpsItem(String dstlpsItem) {
	if (this.dstlps == null) {
	this.dstlps = new ArrayList<String>();
	}
	this.dstlps.add(dstlpsItem);
	return this;
	}
	/**
	* [] IP [] [] [] [] [] [] [] [] [] [] IPV4[]IPV6 [] [] [] [] [] 1 []
	[] IP[] IP [] [] [] IP [] [] [] [] [] []
	* return dstlps
	**/
	@ApiModelProperty(value = "[[] IP [] [] [] [] [] [] [] [] [] [] IPV4[]IPV6 [] [] [] [] [] 1 [] [] IP[] IP [] [] [] IP [] [] [] [] []]")
	public List<String> getDstlps() {
	return dstlps;
	}
	/**
	* [] IP [] [] [] [] [] [] [] [] [] [] IPV4[]IPV6 [] [] [] [] [] 1 []
	[] IP[] IP [] [] [] IP [] [] [] [] [] []
	* Param dstlps
	**/
	public void setDstlps(List<String> dstlps) {
	this.dstlps = dstlps;
	}
	@Override
	public boolean equals(java.lang.Object o) {
	if (this == o) {
	return true;

	}
	if (o == null getClass() != o.getClass()) {
	return false;
	}
	InterdictionDTO interdictionDTO =
	(InterdictionDTO) o;
	return Objects.equals(this.blockId,
	interdictionDTO.blockId) &&
	Objects.equals(this.tenant,
	interdictionDTO.tenant) &&
	Objects.equals(this.producer,
	interdictionDTO.producer) &&
	Objects.equals(this.direction,
	interdictionDTO.direction) &&
	Objects.equals(this.srcIps,
	interdictionDTO.srcIps) &&
	Objects.equals(this.dstIps,
	interdictionDTO.dstIps);
	}
	@Override
	public int hashCode() {
	return Objects.hash(blockId, tenant, producer,
	direction, srcIps, dstIps);
	}
	@Override
	public String toString() {
	StringBuilder sb = new StringBuilder();
	sb.append("class InterdictionDTO {\n");
	sb.append(" blockId:
	").append(toIndentedString(blockId)).append("\n");
	sb.append(" tenant:
	").append(toIndentedString(tenant)).append("\n");
	sb.append(" producer:
	").append(toIndentedString(producer)).append("\n");
	sb.append(" direction:

	".append(toIndentedString(direction)).append("\n");
	sb.append(" srcIps: ").append(toIndentedString(srcIps)).append("\n");
	sb.append(" dstIps: ").append(toIndentedString(dstIps)).append("\n");
	sb.append("}");
	return sb.toString();
	}
	/**
	* Convert the given object to string with each line indented by 4 spaces
	* (except the first line).
	*/
	private String toIndentedString(java.lang.Object o) {
	if (o == null) {
	return "null";
	}
	return o.toString().replace("\n", "\n ");
	}
	}